Installing Step by Step

(Version 0.11)

Contents

This HowTo describes the installation and setup of the “Fingerprint GUI” project. It was tested on *Ubuntu 8.04, 8.10 and 9.04 Desktop* and *Fedora 10* (32bit versions) and *Ubuntu 8.10 and 9.04 Desktop* (64bit version) each new installed with default settings. It is applicable for GDM/Gnome desktop systems only and **can not be used as a HowTo for KDE systems**. It should show the principles of installing and configuring the system and provide enough information needed for deployment in other Linux distributions. In the chapter **"Other Linux distributions"**, my experiences with these distributions are described. I didn't have the time to solve all those problems. That should be the responsibility of the distributors or of experienced users. Please contact me if you have the system installed properly to such a distribution or if you experience a bug.

1 Installing Executables and Libraries

After downloading the “fingerprint-x.y.tar.gz” package please unpack it into some directory (tar -xzf fingerprint-x.y.tar.gz). Then change to this directory and become “root”. The command for installation is “./install.sh [--with-upek | --fprint-only | --uninstall]”. The commandline option “--with-upek” or “--fprint-only” depends on the fingerprint device installed on your system. If you have a device from UPEK Inc. or SGS Thomson you need the “--with-upek” option. This option installs the proprietary “libbsapi.so” library to your system which is used to control these devices. If you have a device from other vendors you can install using the “--fprint-only” option. Please have a look at the libfprint homepage (http://reactivated.net/fprint/wiki/Main_Page) for a list of supported devices.

If you are not sure and can accept installing a proprietary library from UPEK Inc. (http://www.upek.com/solutions/pc_and_networking/sdks/linux/) it's always safe to install the "--with-upek" version.

1.1 Installing required Libraries

When executing "./install.sh" as root it will probably print a list of missing libraries. Use your package manager to install the required packages and their dependencies. Below is a list of packages to install:

Ubuntu 8.10 Desktop:

- libfakekey0
- libfprint0
- libqca-ossl

Fedora 10:

- libfakekey-0.1-1.fc10
- libfprint-0.0.5-6.fc10
- qt-x11-1:4.4.3-10.fc10
- libqca-ossl

There is a special case if you install on a 64bit system and want to use the "--with-upek" option: The proprietary "libbsapi.so" is a 32bit library only. The "install.sh" script takes care of this situation and installs the 32bit binaries of "fingerprintGUI", "fingerprintIdentifier" and "fingerprintHelper" which make use of this library. On Ubuntu it uses "getlibs" (http://www.boundlesssupremacy.com/Cappy/getlibs/) to install the missing 32bit libraries to your system.

1.2 Enabling uinput

The fingerprint PAM module needs "uinput" for proper work. To load the "uinput" module edit your "/etc/modules" file and append a line "uinput" if it isn't there. Restart your system after editing this file.

1.3 Installing executables

If all required libraries are installed the "./install.sh" script will copy the executables to the following locations:

- "fingerprintGUI", "fingerprintIdentifier", "fingerprintHelper" and "fingerprintPlugin" to /usr/local/bin/,
- A "Fingerprint GUI" entry in the "System Settings" menu,
- The library "libpam_fingerprint.so" to /lib/security,
- and in case of "--with-upek" the library "libbsapi.so" to /usr/local/lib.

2 Acquiring Fingerprints

Now you should be able to call "fingerprintGUI" from the command line or use the "Fingerprint GUI" entry in "System Settings" menu. Acquiring fingerprints should be self-explanatory in the fingerprintGUI program. Your fingerprints are stored in a "~/.fingerprints" directory in the user's home directory. If you give the "debug" argument to "fingerprintGUI" a lot of debug output is given to syslog (or /var/log/auth.log).

After some users have registered their fingerprints you can test the fingerprint identification by calling “fingerprintIdentifier” as root. This application can identify your users and print their login names to stdout.

3 Setting up Fingerprint Authentication

You need root permissions to make changes to your PAM configuration. First of all make a copy of your “/etc/pam.d/common-auth” file and name it “/etc/pam.d/common-auth.fingerprint”. Edit this file like follows:

- insert a line `"auth  sufficient  libpam_fingerprint.so  debug"` **as the first line**;
- find the line containing “pam_unix.so” and add the argument “try_first_pass” to the call of “pam_unix.so”;

The distributions differ slightly with regard to the filenames and their contents:

<u>Ubuntu 8.10 Desktop:</u>

“/etc/pam.d/common-auth.fingerprint” is a copy of “/etc/pam.d/common-auth”. The changed lines in question read:

```
"auth sufficient  libpam_fingerprint.so   debug"
```

```
"auth [success=1 default=ignore]    pam_unix.so try_first_pass nullok_secure"
```

<u>Fedora 10:</u>

“/etc/pam.d/common-auth.fingerprint” is a copy of “/etc/pam.d/system-auth-ac”. The changed lines in question read:

```
"auth sufficient  libpam_fingerprint.so debug"
```

```
"auth sufficient  pam_unix.so try_first_pass"
```

If you're finished setting up your “common-auth.fingerprint” file you can setup the services for fingerprint authentication now. It is assumed you have at least one fingerprint registered for your user account and another one for the root user. Also make sure there is set a password for root (sudo passwd root).

<u>**IMPORTANT NOTE:**</u> The following settings can lock access to your system completely if something goes wrong. So please open a secure tty (ctrl-alt-F2) and login as root there. So you're able to undo the changes made in /etc/pam.d/.

3.1 Configuring “su”

Edit the file “/etc/pam.d/su” and change the line “@include common-auth” to “@include common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). Then open a terminal window and call “su”. A password prompt should appear in the terminal <u>**and**</u> the system should open a GUI widget requesting a finger swipe with the message “Authenticating root” in it's status bar. If you can become root by swiping the finger registered for root it works. You should also be able to become root by ignoring this GUI widget and typing root's password at the prompt.

3.2 Configuring “login”

Edit the file “/etc/pam.d/login” and change the line “@include common-auth” to “@include common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). Then change to a secure tty (e.g. ctrl-alt-F3), type the username and press enter. The password prompt should appear along with a message “Type your password or swipe your finger”. You should be able to login with a finger swipe and with typing the password as well.

3.3 Configuring “sudo”

Edit the file “/etc/pam.d/sudo” and change the line “@common-auth” to “@common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). Make sure your login name is in the sudoers file. Then open a terminal window and call “sudo gnome-terminal”. After swiping your finger the gnome-terminal should open with root permissions.

3.4 Configuring “gdm”

In order to be able to login into a desktop session you need to configure your gdm (probably with gdmsetup). Disable “autologin”, “timed login” and “facebrowser”. Then double check you have a root session on a secure tty open (for undoing the changes if something goes wrong). Edit the file “/etc/pam.d/gdm” and change the line “@include common-auth” to “@include common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). If there is a line reading “auth requisite pam_nologin.so” **comment this line out or remove it**. Now logout from your gnome session. The gdm greeter should show a login prompt **and** the GUI widget requesting a finger swipe below. You should be able to login with fingerprint and with name/password as well.

3.5 Configuring “gnome-screensaver”

Gnome-screensaver needs a plugin to display the fingerprint GUI widget to the user while unlocking. To start this plugin with the gnome-screensaver-dialog open the gconf-editor, find the “apps | gnome-screensaver” entry and **enable** the “embedded_keyboard_enabled” item. Then invoke the string “/usr/local/bin/fingerprintPlugin -d” as the “/apps/gnome-screensaver/embedded_keyboard_command” and close gconf-editor. **This step needs to be taken by every user who wants to unlock his/her gnome-screensaver by fingerprint on that machine!**

Then edit the file “/etc/pam.d/gnome-screensaver” change the line “@include common-auth” to “@include common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). Double check you have a root session on a secure tty open (for undoing the changes if something goes wrong) before testing. You can now lock your screen and should be able to unlock it with a fingerswipe or with your password.

4 Password Store

There are applications that need a password for encrypting or decrypting something on your system. Probably gnome-keyring is the most widespread of such applications. These applications sometimes get their key for decrypting (e.g. for the password safe) by querying the PAM session environment for the password given by the user at login. But when the user was logged in with a fingerprint there is no password stored in the PAM session environment. So the application will prompt the user for a password when needed (e.g. if a wireless WPA connection has to be established by the Gnome Network Manager or if you want to access your email account with Evolution) even if he was logged in already.

Since version 0.11 of FingerprintGUI there is a solution: You can use some removable media (USB stick) to save your (encrypted) password there. If the media is connected to your machine while you login with your fingerprint the fingerprintPAM module can decrypt the password and send it to the PAM session environment.

PLEASE READ CAREFULLY NOW AND USE THIS FEATURE ONLY IF YOU UNDESTAND HOW IT WORKS!

If you use the “Password” tab of FingerprintGUI you can chose a directory on some removable media, then type your login password twice and click the “Save” button. The removable media must be mounted and you must have write permission there. This is where FingerprintGUI creates a subdirectory “.fingerprints” and writes a file “<username>@<machinename>.xml” containing the encrypted password there. The key for decrypting this password, the path for the “<username>@<machinename>.xml” file and the UUID of the removable media are saved in a file “~/.fingerprints/config.xml” in your home directory (probably on your local HDD).

When you login using your fingerprint the fingerprintPAM module reads the “~/.fingerprints/config.xml” file, finds the “<username>@<machinename>.xml” file on the removable media (if it is connected and has the given UUID), mounts it, decrypts the password and saves it to the PAM session environment where gnome-keyring or other permitted applications can read it. This avoids your system asking for a password again.

PLEASE NOTE THE FOLLOWING RESTRICTIONS:

- Do not use this feature if someone other then you has root permissions on this machine. This is because root can connect to the machine via telnet, ssh or something like this, mount the external media, find the “<username>@<machinename>.xml” file, read the “~/.fingerprints/config.xml” file in your home directory and decrypt your password.
- Do not connect the removable media if it isn't needed. The FingerprintPAM module only needs it while login is in progress. It mounts the partition with the given UUID containing the “<username>@<machinename>.xml” file and unmounts it immediately after it has read the file.
- Do never leave the removable media and the computer at the same location unattended. Someone could copy both files and decrypt your password later.
- You don't need to type your password any more so you can use a very long and strong password now. But do not forget your password! You would not be able to unlock your login-keyring any more if your removable media gets lost or corrupted.
- If you change your login password on this machine you need to use FingerprintGUI and save the new password to the removable media again.

This is how I use this feature for myself:

My USB stick has 3 partitions: One “vfat” (/dev/sdb1) to keep files to be transferred to other machines, one “luks_crypto” (/dev/sdb2) partition to keep my secret data and a very small (3MB) “ext2” (/dev/sdb3) partition to hold the “<username>@<machinename>.xml” file. Corresponding entries in /etc/fstab ensure that the partitions sdb2 and sdb3 are not automatic mounted. Needless to say that I'm the only person who has root access to my notebook.

While booting my notebook I connect the USB stick until I'm logged in with my fingerprint, then remove the stick immediately and reconnect it only (and only as long as needed!) if I want to copy something from or to it. Because I don't need to invoke my password any more I use a very strong and cryptic login password.

5 Troubleshooting

5.1 FingerprintGUI Error “Could not open fingerprint device”

On some systems the file “/etc/udev/rules.d/40-libfprint0.rules” (or something like this) installed by the “libfprint” package doesn't work properly. It should help to rename this file so it is invoked at a later time. In all know cases renaming it to “91-libfprint0.rules” solved the problem. **You should also make sure your fingerprint scanner hardware has an entry in this file**.

5.2 Login on a secure tty hangs with “OK” Message

If you try to login on a secure tty the prompt “Swipe your finger or type your password” appears. If you swipe the finger the message “OK” appears and then nothing happens. In this case the “uinput” device doesn't work. Make sure the “uinput” module is loaded (“lsmod | grep uinput”), the device exists in “/dev/input/uinput”, “/dev/misc/uinput” or “/dev/uinput” and you have write permission to it.

5.3 Password can not be saved to removable media

If you find an entry reading:

```
"AES128-CBC not supported! Provider (libqca-ossl.so) not installed?"
```

in the log files, the plugin library for encryption is missing. Install the “libqca2-plugin-ossl” package (Ubuntu) or a similar encryption plugin.

In other cases make sure the media is removable, contains a valid partition and is mounted with read/write permission.

6 Known Limitations

6.1 Applications that don't use PAM for prompting a password

The normal way to use PAM for authentication is to let the PAM system prompt the user for a username and/or a password. PAM uses then a callback function of the calling application for prompting something in it's own style. If called back by PAM the application can decide how it wants to prompt for name or password; if not called back, PAM has performed the authentication in another way (fingerprint, smart card, iris scanner or whatever). Maybe they didn't understand that or had another reason not to use that mechanism, the developers of some applications decided to prompt for password or username before calling PAM. In this case the libpam_fingerprint.so library is called at a time where the password is already known by the PAM stack and therefore exits immediately. Fingerprint authentication is not possible then.

6.2 Missing XAUTHORITY environment variable

When calling PAM some applications don't have a XAUTHORITY variable in their environment. Libpam_fingerprint.so tries hard to find the “MIT Magic Cookie” to be used to connect to the current display but in some cases it fails. I guess this is in several KDE applications the reason for not being able to show the fingerprint widget. Maybe I'll find some better solution in a later version.

6.3 Other Linux distributions

Debian 4.0

I didn't find any way to install libfprint. There is neither a package available nor do the sources compile without errors. Didn't want to waste more time with it.

SuSE 11.1 (gnome edition)

The gdm used in SuSE behaves totally strange. It doesn't allow to show the fingerprint widget. Maybe it's only some setting to be changed or the original source installation of gdm to be used. Neither found any useful documentation about it nor had the time to try a fresh compiled gdm from sources. Gave up!

Slackware

Slackware might need someone who has enough spare time to make it “PAM aware”. Not me!

So if you are interested to bring Fingerprint GUI to work on some other distributions first read the “Hacking.pdf” document of this project for hints about how it works. If you need further information about it contact me. If you managed to make it up and running write a HowTo and let me know.

Ubuntu and Fedora users should have no serious problems; so have fun with it!